Setting up MySQL on the Cluster after Rebuild

Samantha Wohlstadter

February 4, 2019

(In Progress)

1 Setting up the Configuraton File

When first starting MySQL, you need to make sure the correct paths to the socket file and the databases are specified in the configuration file. It is located on the CE at `/etc/my.cnf`

The file `/opt/rocks/mysql/my.cnf` should contain the correct information which can simply be copied over to `/etc/my.cnf`, but this file is not looked at by MySQL when it runs. If this file does not exist, the correct socket should be `socket=/var/opt/rocks/mysql/mysql.sock`, and the data directory should be `datadir=/var/opt/rocks/mysql/`. Make sure to specify the paths under both the [mysqld] and the [client] headings.

After making changes to the configuration file, make sure to restart apachectl and the mysql server. `/usr/sbin/apachectl graceful` `/opt/rocks/mysql/support-` `restart`

2 Importing Databases

The databses used in mysql for the cluster tend to be in "InnoDB" format. What that means as far as importing databases go is that one cannot simply copy the folders directly from the old databse to the new one. MySQL cannot read the files this way. Instead, the old databases must be 'dumped' using the mysqldump command.

2.1 The Proper Procedure

To export a database, run the following: `mysqldump -u root -p (database) > (database).sql`

To import a database, first create the database in mysql using ``` ``create database (name); ``` Then, quit MySQL and run the following: `mysql -u root -p (database) < (database).sql`

Once you have imported the databases, be sure to give the appropriate permissions. For example, granting permission for apache to access the wiki-database: `GRANT ALL PRIVILEGES ON (database)* TO (user)@localhost`

Be sure to update appripriate config files when changes are made, such as LocalSettings for the wiki page.

2.2 The Hard Way

If you do not have the .sql files, and the old mysql server is not able to run, the databases can still be accessed to create .sql files. You will need the "ibdata1" file from the old mysql server to be able to open the databases. The old mysql files are located on nas1 in CEBackup-20180803/var/lib/mysql. The following procedure was performed on a different machine than the new mysql server, though it can be done on the same machine.

- Stop MySQL using `/opt/rocks/mysql/support-files/mysql.server stop`
- Create a copy of the current data directory (in this case it is /var/opt/rocks/mysql) and rename it (backup)
- Copy the desired database folders (excluding "mysql") from the old server to the new server.
- Start MySQL using /opt/rocks/mysql/support-files/mysql.server start
- Log in to mysql and run "use (database); show tables; select * from (table);" on the old databases you just copied over, one at a time. You should get "table does not exist". This is okay at this point, you are "showing" mysql that these databases and tables exist in its data directory now, even though it can't read them yet.
- Quit mysql, then stop using /opt/rocks/mysql/support-files/mysql.server stop

- Replace the ibdata1 file with the file of the same name from the old MySQL server.
- (Optional?) Replace the mysql database with the databse from the old server. (You must know the password from the old server to log in).
- Start the mysql service with /opt/rocks/mysql/support-files/mysql.server start
- Log in to mysql and test the database with "use database; show tables; select * from (tablename)" You should get a table of data as the output now.
- If the above step was successful, log out of mysql and use mysqldump as above to create the .sql files and import them as above. If you used the same machine to do this, make sure to rename the current mysql directory to something else, and rename the backup to "mysql/" You should then be able to access the data in the new server. (use (database); show tables; select * from (table);)